Step 4: Apply steps 2, 3, 4, 5 of $\Rightarrow_{\text{ECNF}}$

Remark: The $\Rightarrow_{\text{OCNF}}$ algorithm is already close to a state of the art algorithm. Missing are further redundancy tests and simplification mechanisms we will discuss later on in this section.

2.5 Superposition for PROP(Σ)

Superposition for PROP(Σ) is:

- resolution (Robinson 1965) +
- ordering restrictions (Bachmair & Ganzinger 1990) +
- abstract redundancy critrion (B&G 1990) +
- partial model construction (B & G 1990) +
- partial-model based inference restriction (Weidenbach)

Resolution for PROP(Σ)

A *calculus* is a set of *inference* and *reduction* rules for a given logic (here PROP(Σ)).

We only consider calculi operating on a set of clauses N. Inference rules *add* new clauses to N whereas reduction rules *remove* clauses from N or *replace* clauses by "simpler" ones.

We are only interested in unsatisfiability, i.e., the considered calculi test whether a clause set N is unsatisfiable. So, in order to check validity of a formula ϕ we check unsatisfiability of the clauses generated from $\neg\phi$.

For clauses we switch between the notation as a disjunction, e.g., $P \vee Q \vee P \vee \neg R$, and the notation as a multiset, e.g., $\{P, Q, P, \neg R\}$. This makes no difference as we consider $\vee$ in the context of clauses always modulo AC. Note that $\bot$, the empty disjunction, corresponds to $\emptyset$, the empty multiset.

For literals we write L, possibly with subscript.. If $L = P$ then $\bar{L} = \neg P$ and if $L = \neg P$ then $\bar{L} = P$, so the bar flips the negation of a literal.

Clauses are typically denoted by letters C, D, possibly with subscript.

The *resolution calculus* consists of the inference rules *resolution* and *factoring*:

$$\text{Resolution} \quad \mathcal{I}\,\frac{C_1 \vee P \quad C_2 \vee \neg P}{C_1 \vee C_2} \qquad\qquad \text{Factoring} \quad \mathcal{I}\,\frac{C \vee L \vee L}{C \vee L}$$

where C_1, C_2, C always stand for clauses, all inference/reduction rules are applied with respect to AC of $\vee$. Given a clause set N the schema above the inference bar is mapped to N and the resulting clauses below the bar are then *added* to N.

and the reduction rules *subsumption* and *tautology deletion*:

$$\text{Subsumption} \quad \mathcal{R}\frac{C_1 \quad C_2}{C_1} \qquad \text{Tautology Deletion} \quad \mathcal{R}\frac{C \vee P \vee \neg P}{}$$

where for subsumption we assume $C_1 \subseteq C_2$. Given a clause set N the schema above the reduction bar is mapped to N and the resulting clauses below the bar *replace* the clauses above the bar in N.

Clauses that can be removed are called *redundant*.

So, if we consider clause sets N as states, $\uplus$ is disjoint union, we get the rules

Resolution $(N \uplus \{C_1 \vee P, C_2 \vee \neg P\}) \quad \Rightarrow \quad (N \cup \{C_1 \vee P, C_2 \vee \neg P\} \cup \{C_1 \vee C_2\})$

Factoring $(N \uplus \{C \vee L \vee L\}) \quad \Rightarrow \quad (N \cup \{C \vee L \vee L\} \cup \{C \vee L\})$

Subsumption $(N \uplus \{C_1, C_2\}) \quad \Rightarrow \quad (N \cup \{C_1\})$

provided $C_1 \subseteq C_2$

Tautology Deletion $(N \uplus \{C \vee P \vee \neg P\}) \quad \Rightarrow \quad (N)$

We need more structure than just (N) in order to define a useful rewrite system. We fix this later on.

Theorem 2.11 *The resolution calculus is sound and complete:*
$$N \text{ is unsatisfiable iff } N \Rightarrow^* \{\bot\}$$

Proof. Will be a consequence of soundness and completeness of superposition. □

Ordering restrictions

Let $\prec$ be a total ordering on Σ.

We lift $\prec$ to a total ordering on literals by $\prec \subseteq \prec_L$ and $P \prec_L \neg P$ and $\neg P \prec_L Q$ for all $P \prec Q$.

We further lift $\prec_L$ to a total ordering on clauses $\prec_C$ by considering the multiset extension of $\prec_L$ for clauses.

Eventually, we overload $\prec$ with $\prec_L$ and $\prec_C$.

We define $N^{\prec C} = \{D \in N \mid D \prec C\}$.

Eventually we will restrict inferences to maximal literals with respect to $\prec$.

Abstract Redundancy

A clause C is *redundant* with respect to a clause set N if $N^{\prec C} \models C$.

Tautologies are redundant. Subsumed clauses are redundant if $\subseteq$ is strict.

Remark: Note that for finite N, $N^{\prec C} \models C$ can be decided for PROP(Σ) but is as hard as testing unsatisfiability for a clause set N.

Partial Model Construction

Given a clause set N and an ordering $\prec$ we can construct a (partial) model $N_{\mathcal{I}}$ for N as follows:

$N_C := \bigcup_{D \prec C} \delta_D$

$$\delta_D := \begin{cases} \{P\} & \text{if } D = D' \lor P \text{ and } P \text{ maximal and } N_D \not\models D \\ \emptyset & \text{otherwise} \end{cases}$$

$N_{\mathcal{I}} := \bigcup_{C \in N} \delta_C$

Superposition

The *superposition calculus* consists of the inference rules *superposition left* and *factoring*:

Superposition Left $(N \uplus \{C_1 \lor P, C_2 \lor \neg P\}) \quad \Rightarrow \quad (N \cup \{C_1 \lor P, C_2 \lor \neg P\} \cup \{C_1 \lor C_2\})$

where P is strictly maximal in $C_1 \lor P$ and $\neg P$ is maximal in $C_2 \lor \neg P$

Factoring $(N \uplus \{C \lor P \lor P\}) \quad \Rightarrow \quad (N \cup \{C \lor P \lor P\} \cup \{C \lor P\})$

where P is maximal in $C \lor P \lor P$

examples for specific redundancy rules are

Subsumption $(N \uplus \{C_1, C_2\}) \quad \Rightarrow \quad (N \cup \{C_1\})$

provided $C_1 \subset C_2$

Tautology Deletion $(N \uplus \{C \lor P \lor \neg P\}) \quad \Rightarrow \quad (N)$

Subsumption Resolution $(N \uplus \{C_1 \lor L, C_2 \lor \bar{L}\}) \quad \Rightarrow \quad (N \cup \{C_1 \lor L, C_2\})$

where $C_1 \subseteq C_2$

Theorem 2.12 *If from a clause set N all possible superposition inferences are redundant and $\bot \notin N$ then N is satisfiable and $N_{\mathcal{I}} \models N$.*